ROBERT A. CESARI (1928-2008)
JOHN F. MCKENNA
MARTIN J. O'DONNELL
THOMAS C. O'KONSKI
PATRICIA A. SHEEHAN
MICHAEL E. ATTAYA
CHARLES J. BARBAS
MICHAEL R. REINEMANN
KEVIN GANNON
DUANE H. DREGER
JAMES A. BLANCHETTE
JAMES M. BEHMKE
SHANNEN C. DELANEY
OMAR M. WADHWA
RITA M. ROONEY
MICHAEL T. ABRAMSON
STEPHEN D. LEBARRON

CESARI AND MCKENNA, LLP
ATTORNEYS AT LAW
88 BLACK FALCON AVENUE
BOSTON, MASSACHUSETTS

Telephone: (617) 951-2500 Telecopier: (617) 951-3927
Website: www.c-m.com

INTELLECTUAL PROPERTY AND RELATED CAUSES

A. SIDNEY JOHNSTON
EDWIN H. PAUL
OF COUNSEL

HEATHER SHAPIRO
PATENT AGENT

FACSIMILE COVER SHEET

112056-0474

DATE:	August 17, 2009
TOTAL PAGES WITH COVER:	22
TO:	Giovanna B. Colan
FIRM:	United States Patent and Trademark Office
FACSIMILE NUMBER:	571-273-2752
TELEPHONE NUMBER:	571-272-2752
FROM:	Michael T. Abramson

COMMENTS:

Please call Kristin at 617-951-3089 to confirm receipt of this agenda.

Thank you!

SPECIAL INSTRUCTIONS:

If you do not receive all pages, or you are not the intended recipient, please contact us at (617) 951-2500 as soon as possible.

PATENTS
112056-0474
P01-2475.01

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re The Application of:
Hristo Iankov Bojinov

Serial No.: 10/803,788

Filed: March 17, 2004

For: METHOD AND APPARATUS FOR IMPROVING FILE SYSTEM PROXY PERFORMANCE AND SECURITY BY DISTRIBUTING INFORMATION TO CLIENTS VIA FILE HANDLES

Examiner: Colan, Giovanna B

Art Unit: 2162

Confirmation No.: 8050

Cesari and McKenna, LLP
88 Black Falcon Avenue
Boston, MA 02210
August 17, 2009

AGENDA FOR INTERVIEW

The Agenda is:
(1) Explain the problem solved
(2) Analyze the claimed solution
(3) Analyze all cited art
(4) Explain why all claims are allowable in view of the cited prior art

Present Status of Case

This Agenda for a telephonic interview with Examiner Colan is sent in response to the Office Action mailed by USPTO on July 8, 2009.

- This Agenda for a telephonic interview with Examiner Colan is sent via facsimile:
  - **Facsimile** # (571)-273-2752
  - **Telephone** # (571)-272-2752
- Attorney Michael T. Abramson (Reg. No. 60,320) will call Examiner Colan for the scheduled interview on **August 27, 2009** at 11:00 AM (EST).

PROPOSED CLAIMS:

1. (Proposed Amendment) A method for establishing identity in a file system, comprising:

receiving, from a client, a first Network File System (NFS) operation~~file request~~ concerning an indicated file ~~from a client~~, the first NFS operation~~request~~ received by a proxy;

forwarding the first NFS operation~~request~~ from the proxy to be received by a file server;

returning a NFS file handle~~reply~~ associated with the first NFS operation~~file request~~ from the file server to the proxy in response to the file server receiving the first NFS operation from the proxy;

inserting, by the proxy, metadata into ~~a~~ the NFS file handle in response to receiving the NFS file handle from the file server, wherein the metadata is an encryption key; ~~and~~

sending, by the proxy in response to receiving the NFS file handle from the file server, the NFS file handle with the metadata inserted in the NFS file handle to the client as a reply to the first NFS operation; and

using, by the client, the metadata and the NFS file handle in a second NFS operation ~~to be used in further requests~~ to identify the client and the indicated file.

2. (Previously Presented) The method of Claim 1, further comprising:

using the metadata in the file handles for any of eliminating a need for the proxy to generate additional requests to the server to establish file identity, and for completing client requests.

3. (Previously Presented) The method of Claim 1, further comprising:

encoding metadata in a form of a session key into the file handle, the session key expiring after a predetermined amount of time.

4. (Previously Presented) The method of Claim 1, further comprising:

using an NFS file system as the file system.

5. (Previously Presented) The method of Claim 1, further comprising:
using a stateless protocol by the file system.

6-29. (Cancelled).

30. (Previously Presented) The method of claim 1, further comprising:
receiving, from the client, a second file request by the proxy, the second file request including the metadata in a further file handle sent with the second request;
identifying, in response to the metadata, the client as having a permission to submit the second file request;
sending the second file request to the file server and not sending the metadata with the second file handle to the file server; and
receiving by the proxy the further reply from the file server, and sending by the proxy the further reply to the client.

31. (Previously Presented) A method for establishing identity in a file system, comprising:
receiving a first file request concerning an indicated file from a client, the first file request received by a proxy;
forwarding the first file request from the proxy to a file server;
returning a reply associated with the first file request from the file server to the proxy, wherein the reply includes a file handle associated with the indicated file;
inserting, by the proxy, metadata into the file handle;
sending, by the proxy, the file handle with the metadata inserted in the file handle to the client, the metadata to be used in further requests to identify the client as having a permission to access the indicated file;
receiving, from the client, a second file request by the proxy, the second file request including the metadata in a second file handle sent with the second file request;

identifying, in response to the metadata, that the client has the permission to submit the second file request;

sending the second file request to the file server and not sending the metadata with the second file handle to the file server; and

receiving by the proxy a second reply from the file server, and sending by the proxy the second reply to the client.

32. (Previously Presented) An apparatus to establish identity in a file system, comprising:

a proxy to receive a file request sent by a client to the file system, the proxy to forward the request to a file server;

the file server to return a reply associated with the file request to the proxy, wherein the reply includes a file handle;

the proxy to insert metadata into the file handle; and

the proxy to send the file handle with the metadata inserted in the file handle to the client, the metadata to be used in further requests to identify the client and the indicated file.

33. (Previously Presented) The apparatus as in claim 32, further comprising:

the proxy to receive, by the client, a second file request, the second file request to include the metadata in the second file handle sent with the second request;

the proxy to identify, in response to the metadata, the client as having a permission to submit the second file request;

the proxy to send the second file request to the file server and not to send the metadata with the second file handle to the file server; and

the proxy to receive a second reply from the file server, and the proxy to send the second reply to the client.

34. (Previously Presented) The apparatus of Claim 32, further comprising:

the proxy to use the metadata in the file handle received from the client to eliminate a need for additional communication with the file server to establish file identity.

35. (Previously Presented) The apparatus of Claim 32, further comprising:
the proxy to encode the metadata in a form of a session key into the file handle, the session key expiring after a predetermined amount of time.

36. (Previously Presented) The apparatus of Claim 32, further comprising:
an NFS file system used as the file system.

37. (Previously Presented) The apparatus of Claim 32, further comprising:
a stateless protocol used by the file system.

38. (Previously Presented) A non-volatile memory executed on a computer, comprising:
said non-volatile memory containing procedures for execution on the computer for a method of establishing identity in a file system, the method having the steps of;
receiving a file request concerning an indicated file from a client, the request received by a proxy;
forwarding the request from the proxy to a file server;
returning a reply associated with the file request from the file server to the proxy, wherein the reply includes a file handle associated with the indicated file;
inserting, by the proxy, metadata into the file handle; and
sending, by the proxy, the file handle with the metadata inserted in the file handle to the client, the metadata to be used in further requests to identify the client and the indicated file.

39. (Previously Presented) A method for establishing identity in a file system, comprising:

receiving a first file request concerning an indicated file from a client, the first file request received by a proxy;

forwarding the first file request from the proxy to a file server;

granting a permission for the request to be acted upon by the file system in response to a predetermined protocol;

returning a reply associated with the first file request from the file server to the proxy, wherein the reply includes a file handle associated with the indicated file;

inserting, by the proxy, a session key into the file handle; and

sending, by the proxy, the file handle with the session key inserted in the file handle to the client, the session key to be used in further requests to identify the client and the indicated file.

40. (Previously Presented) The method according to claim 39, further comprising:

receiving, from the client, a second file request by the proxy, the second file request including the session key in a second file handle sent with the second file request;

identifying, in response to the session key, that the client has the permission to submit the second file request;

sending the second file request to the file server and not sending the session key with the second file handle to the file server; and

receiving by the proxy a second reply from the file server, and sending by the proxy the second reply to the client.

41. (Previously Presented) The method according to claim 39, further comprising:

causing the session key to expire after a selected amount of time.

42. (Previously Presented) The method according to claim 39, further comprising:

causing the session key to expire after a selected amount of usage.

43. (Previously Presented) The method according to claim 39, further comprising:

using a NFS protocol as the predetermined protocol.

44. (Previously Presented) The method according to claim 43, further comprising:
using as the predetermined protocol a two way communication exchange between the proxy and the file server.

45. (Previously Presented) An apparatus to establish identity in a file system, comprising:
a proxy to receive a file request sent by a client to the file system, the proxy to forward the request to a file server;
the file server to return a reply associated with the file request to the proxy, wherein the reply includes a file handle;
the proxy to insert a session key into the file handle; and
the proxy to send the file handle with the session key inserted in the file handle to the client, the session key to be used in further requests to identify the client and the indicated file.

46. (Previously Presented) The apparatus as in claim 45, further comprising:
the proxy to receive, by the client, a second file request, the second file request to include the session key in a further file handle sent with the second request;
the proxy to identify, in response to the session key, the client as having a permission to submit the another file request;
the proxy to send the second request to the file server and not to send the session key with the second file handle to the file server; and
the proxy to receive a further reply from the file server, and the proxy to send the further reply to the client.

47. (Previously Presented) The apparatus of Claim 45, further comprising:
the proxy to use the metadata in the file handle received from the client to eliminate a need for additional communication with the file server to establish file identity.

48. (Previously Presented) The apparatus of Claim 45, further comprising:
the proxy to encode the metadata in a form of a session key into the file handle, the session key expiring after a predetermined amount of time.

49. (Previously Presented) The apparatus of Claim 45, further comprising:
an NFS file system used as the file system.

50. (Previously Presented) The apparatus of Claim 45, further comprising:
a stateless protocol used by the file system.

51. (Previously Presented) An apparatus to establish identity in a file system, comprising:
a proxy configured to receive a first file request sent by a client to the file system, the proxy further configured to forward the first file request to a file server;
the file server configured to return a reply associated with the first file request to the proxy;
the proxy further configured to insert a session key into a file handle;
the proxy further configured to send the file handle with the session key inserted in the file handle to the client, the session key configured to be used in a second file request to identify the client and the indicated file;
the proxy further configured to receive, by the client, a second file request, the second file request configured to include the session key in a second file handle sent with the second file request;
the proxy further configured to identify, in response to the session key, the client as having a permission to submit the second file request;
the proxy further configured to send the second file request to the file server and not to send the session key with the second file handle to the file server; and
the proxy further configured to receive a second reply from the file server, and the proxy further configured to send the second reply to the client.

52. (Previously Presented) A method for establishing identity in a file system, comprising:

receiving a first file request concerning an indicated file from a client, the first file request received by a proxy;

forwarding the first file request from the proxy to a file server;

determining that the client has a permission to have the request acted upon by the file system in response to a predetermined protocol;

returning a reply associated with the first file request from the file server to the proxy, wherein the reply includes a file handle associated with the indicated file;

inserting, by the proxy, a cryptographic information into the file handle;

sending, by the proxy, the file handle with the cryptographic information inserted in the file handle to the client, the cryptographic information to be used in one or more requests to identify the client and the indicated file.

53. (Previously Presented) The method according to claim 52, further comprising:

receiving, by the client, a second file request by the proxy, the second file request including the cryptographic information in a second file handle sent with the second file request;

identifying, in response to the cryptographic information, that the client has the permission to submit the second file request;

sending the second file request to the file server and not sending the cryptographic information with the second file handle to the file server; and

receiving by the proxy a second reply from the file server, and sending by the proxy the second reply to the client.

54. (Previously Presented) The method according to claim 52, further comprising:

causing the cryptographic information to expire after a selected amount of time.

55. (Previously Presented) The method according to claim 52, further comprising:

causing the cryptographic information to expire after a selected amount of usage.

56. (Previously Presented) The method according to claim 52, further comprising:
using a NFS protocol as the predetermined protocol.

57. (Previously Presented) The method according to claim 52, further comprising:
using as the predetermined protocol a two way communication exchange between the proxy and the file server.

58. (Previously Presented) An apparatus to establish identity in a file system, comprising:
a proxy configured to receive a file request for an indicated file sent by a client to the file system, the proxy further configured to forward the request to a file server;
the file server configured to return a reply associated with the file request to the proxy, wherein the reply is configured to include a file handle;
the proxy further configured to insert a cryptographic information into the file handle; and
the proxy further configured to send the file handle with the cryptographic information inserted in the file handle to the client, the cryptographic information configured to be used in further requests to identify the client and the indicated file.

59. (Previously Presented) The apparatus as in claim 58, further comprising:
the proxy further configured to receive, by the client, a second request, the second file request to include the cryptographic information in a second file handle sent with the second request;
the proxy further configured to identify, in response to the cryptographic information, the client as having a permission to submit the second file request;
the proxy further configured to send the second request to the file server and not to send the cryptographic information with the second file handle to the file server; and
the proxy further configured to receive a further reply from the file server, and the proxy to send the further reply to the client.

60. (Previously Presented) The apparatus of claim 58, further comprising:
the proxy further configured to use the metadata in the file handle received from the client to eliminate a need for additional communication with the file server to establish file identity.

61. (Previously Presented) The apparatus of claim 58, further comprising:
the proxy further configured to encode the metadata in a form of a cryptographic information into the file handle, the cryptographic information configured to expire after a predetermined amount of time.

62. (Previously Presented) The apparatus of claim 58, further comprising:
an NFS file system used as the file system.

63. (Previously Presented) The apparatus of claim 58, further comprising:
a stateless protocol used by the file system.

64. (Previously Presented) An apparatus to establish identity in a file system, comprising:
a proxy configured to receive a first file request sent by a client to the file system, the proxy to forward the first file request to a file server;
the file server configured to return a reply associated with the first file request to the proxy;
the proxy further configured to insert a cryptographic information into a file handle;
the proxy further configured to send the file handle with the cryptographic information inserted in the file handle to the client, the cryptographic information configured to be used in a second file request to identify the client and the indicated file;

the proxy further configured to receive, by the client, a second file request, the second file request configured to include the cryptographic information in a second file handle sent with the second file request;

the proxy further configured to identify, in response to the cryptographic information, the client as having a permission to submit the second file request;

the proxy further configured to send the second file request to the file server and not to send the cryptographic information with the second file handle to the file server; and

the proxy further configured to receive a second reply from the file server, and the proxy to send the second reply to the client.

65. (Previously Presented) A method for establishing identity in a file system, comprising:

receiving a file request concerning an indicated file from a client, the request received by a proxy;

forwarding the request from the proxy to a file server;

returning a reply associated with the file request from the file server to the proxy, wherein the reply includes a file handle associated with the indicated file;

inserting, by the proxy, metadata into the file handle; and

sending, by the proxy, the file handle with the metadata inserted in the file handle to the client, a size of the file handle set to a sum of a length of the server file handle and a length of the proxy metadata, the metadata to be used in further requests to identify the client and the indicated file.

66. (Previously Presented) A method, comprising:

receiving, by a proxy, a file request for a file sent from a client;

forwarding the file request from the proxy to a file server;

returning a reply associated with the file request from the file server to the proxy, wherein the reply includes a file handle;

inserting, by the proxy, metadata into the file handle;

sending, by the proxy, the file handle with the metadata inserted in the file handle to the client; and

using, by the client, the metadata inserted into the file handle in a subsequent file request to identify the client and the file.

67. (Previously Presented) A computer apparatus, comprising:

a proxy configured to receive a client file request for a file and forward the file request from the proxy to a file server;

the server configured to return a reply associated with the file request, wherein the reply includes a file handle;

the proxy further configured to intercept the file handle sent from the server and insert metadata into the file handle to create a modified file handle;

the proxy further configured to send the modified file handle with the metadata inserted in the file handle to the client; and

the proxy further configured to receive the modified file handle from the client for a second file request for the file, wherein the proxy is further configured to use the modified file handle to eliminate a need for the proxy to generate one or more additional requests to the server that would be required to access the file if the modified file handle did not include the inserted metadata.

REMARKS

Claims 1-5 and 30-67 are in the case.

No new claims have been added.

A proposed amendment, in particular to claim 1, is included for discussion.

PROBLEM SOLVED

Network File System (NFS) supports a lookup procedure, which converts a filename into a file handle. This file handle is a unique, immutable identifier, usually an inode number, or disk block address. NFS does have a read procedure, but the client must specify a file handle and starting offset for every call to read. A software program or appliance that is a proxy for the NFS protocol, or any other protocol that uses server-generated file handles, usually requires additional file metadata information to be stored either on the server or locally on the proxy. This metadata can be used, for example, to apply different encryption keys, or to enforce access restrictions to files that are located in different logical units that are defined on the proxy, but possibly invisible to the file server.

Such an appliance forwards file handles generated by the file server to clients, and subsequently acts as a proxy for client requests for access to the file system on the server. Given a file handle from a client, the appliance needs to establish to what area (a.k.a. storage vault) the file belongs, and use the appropriate keys to encrypt or decrypt data. If the metadata used to establish this are not available on the proxy, as is typically the case with large file sets accessed by many client machines, the proxy must send additional requests to the file server to determine how to handle the client request correctly.

It would be advantageous to provide a mechanism that distributes information, effectively caches information, and provides a mechanism that improves performance by eliminating the need for the proxy to generate additional requests to the server to establish file identity.

PATENTS
112056-0474
P01-2475.01

Rejections Under 35 U.S.C. § 103

At Paragraph 6 of the Office Action, claims 1-5 and 30-67 were rejected under 35 U.S.C. § 103(a) as being anticipated by Chandrashekhar et al., U. S. Patent Publication 2005/0033988 published on February 10, 2005 (hereinafter "Chandrashekhar"), and in view of Ryuutou et al., U.S. Patent Application Publication No. 2002/0083191 published on June 27, 2002 (hereinafter "Ryuutou").

Applicant's claimed novel invention, as set out in representative claim 1, comprises in part:

> 1. A method for establishing identity in a file system, comprising:
> receiving, from a client, a first Network File System (NFS) operation concerning an indicated file, the first NFS operation received by a proxy;
> forwarding the first NFS operation from the proxy to be received by a file server;
> **returning a *NFS file handle* associated with the first NFS operation from the file server to the proxy** in response to the file server receiving the first NFS operation from the proxy;
> **inserting, by the proxy, metadata into the NFS file handle** in response to receiving the NFS file handle from the file server, **wherein *the metadata is an encryption key***;
> **sending**, by the proxy in response to receiving the NFS file handle from the file server, **the NFS file handle with the metadata inserted in the NFS file handle to the client *as a reply to the first NFS operation***; and
> using, by the client, the metadata and the NFS file handle in a second NFS operation to identify the client and the indicated file.

Chandrashekhar discusses processing file requests sent by a client and received by a proxy using security applications to encrypt, decompress, verify, and decrypt network data by a server receiving the files from the proxy [0058; 0071]. Header policy information is determined, generated, and then stored on the filer server [0055; Fig. 4-5]. Chandrashekhar states that *the metadata relates to key management, length of the original file/dataset, whether the file was compressed prior to encryption or not, and*

integrity checks for file data [0038]. However, any metadata added to a file is stripped off before the file data/file attributes are returned to the client [0038].

Ryuutou discloses, in relevant part as cited by Examiner, a client establishing an HTTP connection between the client and a proxy server by initiating a communication connection request [0072-0073]. A session ID is added to header information of an HTTP request to determine whether or not a connection corresponding to a particular series of communications have been established [Abstract; 0017; see also Fig. 10 below for an example of an HTTP header information format]. This information about the client is stored in a memory table on the proxy server [0057].

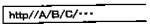

FIG. 10

Applicant respectfully urges that Chandrashekhar, taken singly or in any combination with Ryuutou, does not disclose Applicant's claimed novel use of

returning a *NFS file handle* associated with the first NFS operation from the file server to the proxy;

inserting, by the proxy, metadata into the NFS file handle, wherein *the metadata is an encryption key*; and

sending the NFS file handle with the metadata inserted in the NFS file handle to the client *as a reply to the first NFS operation*.

Applicant claims, in part, a proxy receiving from a client ***a first Network File System (NFS) operation*** concerning an indicated file and forwarding the first NFS operation from the proxy to be received by a file server. Applicant further claims **returning a *NFS file handle* associated with the first NFS operation from the file server to the proxy** in response to the file server receiving the first NFS operation from

the proxy. Applicant further claims **inserting, by the proxy, metadata into the NFS file handle, wherein *the metadata is an encryption key***. With that being said, after inserting (the encryption key) metadata into the ***NFS file handle***, Applicant further claims **sending the NFS file handle with the metadata inserted in the NFS file handle to the client *as a reply to the first NFS operation***.

Applicant respectfully argues that Chandrashekhar does not teach or suggest Applicant's claimed novel **returning a *NFS file handle* associated with the first NFS operation from the file server to the proxy**. It should first be noted that while Chandrashekhar does disclose adding header information on a file by file basis, there is no indication that Chandrashekhar shows or suggests adding header information to a file handle. More to the point, Chandrashekhar does not teach or suggest the concept of a file handle. However, even if it is assumed *arguendo* that Chandrashekhar does disclose a file handle, Chandrashekhar is still silent to a Network File System (***NFS***) file handle. In contrast, Applicant claims returning a ***NFS file handle*** associated with a first NFS operation from the file server to the proxy. As such, even if it is assumed *arguendo* that Chandrashekhar does disclose a file handle, because Chandrashekhar is silent to the concept of a ***NFS*** file handle, Chandrashekhar must still be silent to Applicant's claimed novel **returning a *NFS file handle* associated with the first NFS operation from the file server to the proxy.**

Additionally, even if it is assumed *arguendo* that Chadrashekhar shows an NFS file handle, and the fact that Chadrashekhar strips off any metadata added to a file before the file data/file attributes are returned to the client, Chadrashekhar explicitly states that the metadata added on a file by file basis is "*key management, length of the original file/dataset, whether the file was compressed prior to encryption or not, and integrity checks for file data.*" In contrast, Applicant claims inserting metadata into the NFS file handle, wherein ***the metadata is an encryption key***. As such, because "*key management, length of the original file/dataset, whether the file was compressed prior to encryption or not, and integrity checks for file data*" are not encryption keys, even if it is assumed

arguendo that Chadrashekhar shows an NFS file handle, Chadrashekhar must still be silent to Applicant's claimed novel **inserting, by the proxy, metadata into the NFS file handle, wherein *the metadata is an encryption key***.

Applicant respectfully argues that Ryuutou does not teach or suggest Applicant's claimed novel **sending a NFS file handle with the metadata inserted in the NFS file handle to the client *as a reply to the first NFS operation***. It should first be noted that while Ryuutou shows adding a session ID to header information which is sent back to the client, similarly to Chadrashekhar, Ryuutou is silent to adding the session ID to a file handle or an NFS file handle. However, even if it is assumed *arguendo* that Ryuutou shows adding a session ID to a file handle or NFS file handle, Ryuutou must still be silent to Applicant's claimed sending a NFS file handle with the metadata inserted in the NFS file handle to the client ***as a reply to the first NFS operation***.

More particularly, while Ryuutou shows a client establishing an HTTP protocol communication connection request with a proxy server, Ryuutou shows adding a session ID to header information in response to the HTTP protocol communication connection request which is not an NFS operation. In contrast, Applicant claims sending a NFS file handle with the metadata inserted in the NFS file handle to the client ***as a reply to the first NFS operation***. As such, because Ryuutou is silent to the concept of NFS or an NFS operation (since Ryuutou is discussed in terms of responding to an HTTP protocol communication connection request which is not an ***NFS operation***), even if it is assumed *arguendo* that Ryuutou shows adding a session ID to a file handle, Ryuutou must still be silent to Applicant's claimed **NFS file handle** or **sending a NFS file handle with the metadata inserted in the NFS file handle to the client *as a reply to the first NFS operation***.

Furthermore, Ryuutou does not teach or suggest Applicant's claimed **inserting, by the proxy, metadata into the NFS file handle, wherein *the metadata is an encryption key***. Specifically, Ryuutou shows adding a session ID as header information which is sent back to the client. In contrast, Applicant claims inserting metadata into the

NFS file handle, wherein ***the metadata is an encryption key***. As such, because a session ID is not the same as an encryption key, Ryuutou is also silent to Applicant's claimed novel **inserting, by the proxy, metadata into the NFS file handle, wherein *the metadata is an encryption key*.**

Accordingly, Applicant respectfully urges that Chandrashekhar, taken singly or in any combination with Ryuutou, is legally insufficient to render the presently claimed invention obvious under 35 U.S.C. § 103. Chandrashekhar and Ryuutou, taken singly or in any combination, does not disclose Applicant's claimed

returning a *NFS file handle* associated with the first NFS operation from the file server to the proxy;

inserting, by the proxy, metadata into the NFS file handle, wherein *the metadata is an encryption key*; and

sending the NFS file handle with the metadata inserted in the NFS file handle to the client *as a reply to the first NFS operation*.

Applicant's Interpretation of the Prior Art

Applicant's interpretation of the prior art references was derived, in part, from the following excerpts:

Chandrashekhar

> [0038]...The meta-data relates to key management, length of the original file/dataset, whether the file was compressed prior to encryption or not, integrity checks for file data. The meta-data is stripped off before the file data/file attributes are returned to the client... (emphasis added)

Ryuutou

> [0017] A communication distribution controlling method according to a first preferred embodiment of the present invention is a communication distribution controlling method distributing one communication to any of a plurality of relay devices, which can relay the one communication, in correspondence with a connection request of the one communication within a series of communications from a client. With this method, a communication connection request is received from a client, whether or not a communication connection corresponding to a series of

communications is established is determined according to an identifier written in the communication connection request, and the requested communication is connected to a particular relay device as a relay destination of an established communication connection, if the communication connection is established. (emphasis added)

[0057] FIG. 6 is a flowchart showing the process of a communication connection management method in this preferred embodiment. In FIG. 5, when a new communication connection to a gateway is established in correspondence with the initial communication connection request within one session, and a session ID is set, its contents are stored in a memory (table) not shown. At the same time, a timer not shown is started, and its elapsed time is monitored. (emphasis added)

[0072] As explained with reference to FIG. 9, the session number S4, and the session ID ZZZ are set by the proxy in correspondence with this communication connection request. The newly set session ID is added to the header information, for example, within the reply (1) to the PC-A 31a in FIG. 4, and returned from the proxy 32a to the client side. (emphasis added)

[0073] At this time, ZZZ as the session ID is added between B and C in the header information shown in FIG. 10. As a method adding a session ID, a method such as Netscape Cookie, with which a browser side can recognize and store, for example, data that is additionally described in an HTTP header, is used. (emphasis added)

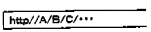

FIG. 10

[0074] A reply including header information to which a session ID is added is returned from a proxy side to a PC side as described above, so that header information including the session ID can be used as the header information in the second and subsequent communication connection requests. (emphasis added)

Conclusion

All newly proposed claims and proposed claim amendments are believed to be fully supported by Applicant's specification.

All proposed independent claims are believed to be in condition for allowance.

All proposed dependent claims are believed to be dependent from allowable proposed independent claims, and therefore in condition for allowance.